Performance of Optical Flow Models on Real-World Occluded Regions

Iris Petre[1]
Supervisor(s): Jan van Gemert [1]**, Sander Gielisse**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Iris Petre
Final project course: CSE3000 Research Project
Thesis committee: Jan van Gemert, Sander Gielisse, Alexios Voulimeneas

Abstract

Occlusions are one of the main challenges in optical flow estimation, where parts of the scene are no longer visible between consecutive frames. Several models address this problem, either intrinsically or explicitly, using different strategies. However, most benchmarks rely on synthetic data, and even real-world ones evaluate only overall model performance, without isolating occlusions. This work investigates optical flow model performance under real-world occlusions by introducing a manually annotated, occlusion-focused dataset. We present an annotation method tailored to three occlusion types: out-of-frame, inter-object, and self-occlusion. We then evaluate two models, FlowFormer++ and CCMR, which handle occlusions using different mechanisms. Our findings show that while CCMR demonstrates stronger overall performance, both models struggle with occluded regions, particularly self-occlusions involving rotation and perspective transformations. These results highlight the need for improved occlusion reasoning in models and more diverse real-world benchmarks.

1. Introduction

Optical flow estimation is the task of predicting apparent motion of objects between pairs of images and is used in medical applications [5], recognition [16], object detection [11] and robotics [2]. A persistent challenge for optical flow models is handling occlusions, where parts of a scene become temporarily hidden between frames. Occlusions often cause prediction errors because the motion cannot be inferred from local visual information alone.

1.1. Existing models

Optical flow models have evolved significantly, from classical methods to deep learning-based architectures capable of capturing complex motion patterns. Despite these advances, occlusion remains one of the most constant challenges in flow estimation [15]. Early convolutional models struggled with occluded regions due to their reliance on local matching [14]. More recent approaches, particularly those based on transformers and attention mechanisms, such as GMA [7], FlowFormer++ [13] [13], and CCMR [6], have introduced global context aggregation or multi-scale reasoning, such in the case of CCMR [6], to better handle occlusions. These models aim to reason about motion even in the absence of direct visual correspondence. However, most existing benchmarks rely on synthetic data, rather than real-world scenes, which limits their ability to reflect true performance in real-world occlusion scenarios. For instance, a rendered box in synthetic data typically has perfect edges and uniform texture. In contrast, a real-world box may have worn corners and soft edges, making occlusion boundaries harder to detect. Moreover, no benchmark specifically focuses on occluded regions, meaning that confounding factors are not isolated and the reported results reflect overall performance rather than occlusion-specific accuracy. This motivates a closer investigation into their actual performance under occlusion-specific conditions.

1.2. Existing datasets

Optical flow models are commonly pre-trained on synthetic datasets such as FlyingChairs [3] and FlyingThings3D [9]. These datasets provide dense and accurate ground truth flow maps, which are challenging to obtain from real-world data. After pretraining, models are often fine-tuned on a combination of synthetic and real-world datasets to enhance generalization [6]. For instance, the KITTI [10] dataset, which includes real-world driving scenarios, is frequently used for fine-tuning. The KITTI [10] dataset includes both non-occluded and occluded ground truth flow maps. However, a noticeable fraction of pixels in the maps with occlusions are marked as invalid, indicating regions where no reliable ground truth could be established, as seen in Figure1. For instance, in the example shown, the red car, representing an out-of-frame occlusion area and the white car, denoting inter-object occlusion, are entirely unannotated, with no valid pixels present in the flow mask. While some occluded or out-of-frame objects are annotated in other scenes, this is not a focus point. Additionally, most scenes depict forward motion, with limited camera rotation or turning. As a result, challenging occlusion cases might be often underrepresented in benchmarks, potentially hiding model performance gaps in real-world scenarios.

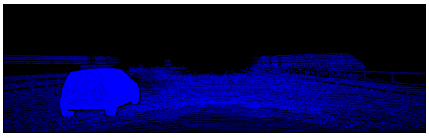
1.3. Research question

The aim of this work is to investigate how optical flow models perform in real-world scenarios involving occlusions. To support this, we introduce a manually annotated dataset focused specifically on occluded regions. Additionally, we evaluated two models, FlowFormer++ [13] and CCMR [6], under different types of occlusions, such as out-of-frame, self-occlusion, and inter-object occlusion, to discover strengths and limitations that may not be evident through the existing benchmarks.

This work makes four key contributions: (1) the development of a custom annotation tool tailored for labelling frame pairs in occluded scenes, (2) a manual annotation method to accurately label different types of occlusions, (3) the evaluation of two optical flow models under occluded regions, and (4) a qualitative and quantitative analysis of the selected model performance across distinct occlusion types. Together, these contributions provide a practical annotation framework and new insights into the limitations of current models when confronted with real-world occlusions.

(a) Frame 00188_10 from KITTI [10], showing the initial scene with multiple vehicles.

(b) Frame 00188_11, the next consecutive frame used for optical flow estimation.



(c) KITTI [10]'s annotated pixels, marked in blue from the flow mask. The silver car is densely annotated, while the other cars have no valid pixels marked, indicating that out-of-frame and inter-object occlusions were not labelled for them.

Figure 1. Example scene from KITTI [10] showing limited occlusion coverage. Despite the presence of multiple moving cars, valid flow labels are provided only for the one in the foreground. This sparsity may underrepresent challenging occlusion scenarios in benchmarks.

2. Related Work

Classical optical flow methods such as Horn and Schunck [4] and Lucas and Kanade [8] introduced foundational techniques based on the brightness constancy assumption, meaning that a pixel's intensity remains constant between two consecutive frames. With the rise of deep learning, convolutional models like RAFT [14] achieved strong performance on benchmarks such as KITTI [10] and Sintel [1] by using convolutional networks to match local image regions and iteratively refine flow predictions, without strictly relying on the brightness constancy assumption. Multi-frame models such as VideoFlow [12] have used temporal information from three or more consecutive frames to improve optical flow estimation, particularly for dealing with occlusions. VideoFlow [12] benefits from richer temporal context but deviates from the widely adopted two-frame setting, which is the focus of our study.

Attention-based methods like GMA [7] introduced transformer-inspired motion aggregation to capture global dependencies. Our paper focuses on two leading models, FlowFormer++ [13] and CCMR [6]. FlowFormer++ [13] enables the model to capture long-range dependencies and effectively reason about occluded regions. Meanwhile, CCMR [6] uses its coarse-to-fine strategy to improve performance in both occluded and non-occluded regions [6]. Both models explicitly target occluded and non-occluded regions, making them suitable for our comparative evaluation.

3. Approach

3.1. Dataset Collection

This work introduces a new real-world dataset focused on occluded scenes, identifying occlusions into three types: self-occlusion, when a part of an object becomes invisible in the second frame due to perspective transformation; inter-object occlusion, when an object is partially hidden by another object in the second frame and out-of-frame occlusion, when (a part of) the object leaves the scene.

The dataset targets sparse annotations, as dense optical flow is impractical to obtain manually, especially in fully occluded regions where accurate labelling is nearly impossible.

An analysis of the KITTI [10] dataset which focuses on urban traffic scenes, revealed that out-of-frame and inter-object occlusions are the most frequent, while self-occlusions occur rarely. However, even the common occlusion types are often excluded from KITTI [10]'s flow masks due to the lack of reliable labels, as seen in Figure1.

To address this gap, the new dataset includes both indoor and outdoor scenes, each featuring examples of one or more occlusion types. While scenes were selected to minimize lighting changes, motion blur, repetitive patterns, and to primarily feature rigid objects, real-world environments naturally include a mix of these challenges. Focusing exclusively on occlusions while eliminating all other factors would create an overly narrow and artificial subset. Therefore, the dataset includes scenes that, while centered around occlusions, still reflect the inherent complexity of real-world scenarios. Textureless surfaces, for example, are difficult to avoid entirely at the pixel level, but the annotation process prioritized well-defined edges and sharp corners wherever possible. All videos were recorded with a smartphone camera at a resolution of 1920×1088 and 30 frames per second.

3.2. Annotation Process

The objective of the annotation process is to generate sparse optical flow ground truths for occluded regions. Manual annotation of hidden points is inherently difficult, even for human annotators, because identifying the correct location of an occluded point requires "anchoring" its surroundings. To address this challenge, we developed an annotation pipeline and used supporting tools.

3.2.1 Annotation Pipeline

To ensure structure and reproducibility, we define a high-level annotation pipeline covering all stages from data collection to evaluation. The process begins with collecting

Figure 2. Out-of-frame annotation example showing how pixels outside KITTI [10]'s resolution boundary are labelled.

videos from various environments, followed by frame extraction and selection using our custom annotation tool. Annotation methods are then applied based on the occlusion type and specific scene characteristics. Once annotations are complete, they are exported to KITTI [10] format and used for both quantitative and qualitative evaluation. The full pipeline is illustrated in Figure 3.

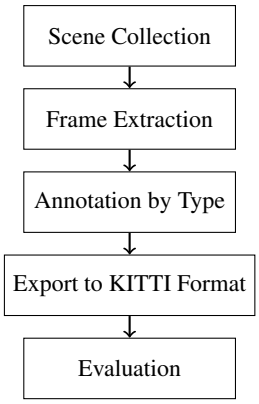


Figure 3. Annotation pipeline, showing the annotation process from data collection to evaluation.

3.2.2 Annotation Tool

An annotation tool was developed in collaboration with the "Real-world Evaluation of Optical Flow" group[1]. The tool allows users to import a video, select frames using a slider, and annotate by clicking corresponding pixels in the two frames. It supports multiple annotation pairs per frame, provides color-coded visual feedback, and exports annotated data in KITTI [10]-compatible format, including the flow mask. This helps with direct model evaluation using the new dataset.

To support occlusion-specific annotations, we implemented several custom features[2], including frame export before annotation, the ability to enter pixel coordinates for occluded points in addition to clicking on the image and visual annotation support for out-of-frame occlusions.

The tool does not annotate the type of occlusion; it only stores the optical flow information. Each frame pair, however, is labelled with one occlusion type to allow evaluation per category. Some frame pairs are annotated multiple times to reflect different types.

3.2.3 Out-of-frame occlusion annotation

To annotate points that leave the visible scene in the second frame, visual support of KITTI [10]'s resolution (1242×375) is used within the higher-resolution video, as shown in Figure 2. A visible rectangle marks this resolution boundary. Points outside this rectangle in the second frame are treated as out-of-frame occlusions. These points are still visible in the full-resolution image and can be annotated. Upon export, the frames are cropped to the KITTI [10] format, preserving the flow for the occluded points. We chose to crop the frames because models support KITTI's [10] resolution, reducing the risk of unexpected runtime errors.

3.2.4 Inter-object annotation

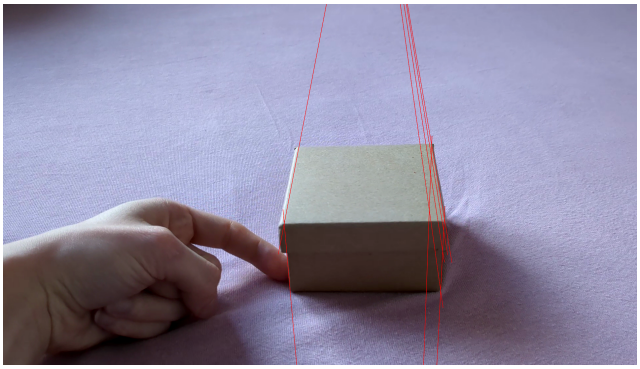
For inter-object occlusions, we adopted what we refer to as the Line Intersection Method. After exporting the relevant frames using our annotation tool, we used Photopea[3], an open-source, web-based image editor, to help with the process. By drawing lines along the visible edges of objects, we marked the intersection points that correspond to the (non)occluded features in both frames, for example, corners, as shown in Figure 4a. These intersection points are

[1] https://github.com/IrisPetre99/RP3000/tree/main

[2] https://github.com/IrisPetre99/RP3000/tree/occlusions

[3] https://www.photopea.com/

(a) Inter-object occlusion annotation example illustrating how occluded areas are marked at intersections.



(b) Self-occlusion annotation example showing the geometrical construction of vanishing points.

Figure 4. Examples of annotations using the Line Intersection Method to mark the intersections of occluded edges.

then annotated by entering their pixel coordinates using the annotation tool. Multiple such points can be used for annotation from a single scene.

3.2.5 Self-occlusion annotation

Self-occlusions, where an object rotates or changes perspective, present a greater challenge. These cases are difficult to annotate due to the effects of 3D-to-2D projection, where lines that are parallel in 3D space no longer appear parallel in the 2D image. This distortion complicates the accurate inference of occluded corners and edges. For example, in a picture with a rotated cube, it is difficult to infer the position of occluded edges by simply translating the visible ones.

We initially estimated the homography matrix from four visible point mappings to infer the occluded point via projective transformation. However, this approach proved unreliable due to its assumption that all selected points lie on the same physical plane in 3D space. The dataset contains moving objects and natural scenes, where it is difficult to isolate surfaces and decompose the scene into multiple planes. Moreover, occluded regions introduce depth discontinuities, making it unreliable to use homography in such areas.

Instead, we extended the Line Intersection Method by incorporating vanishing points. As illustrated in Figure 4b, we first identified vanishing points based on visible edges. Then, by drawing a line from a visible corner through a vanishing point, we could infer the position of an occluded corner. This provided a structured, geometry-based strategy for annotating self-occluded regions.

Annotation difficulty varies on a case-by-case basis. In some scenes, objects, like chairs, have external parts that remain visible despite partial occlusion, allowing for straightforward annotation using the Line Intersection Method without needing to estimate vanishing points. In contrast, for more compact or uniformly shaped objects, accurately inferring the occluded regions becomes more challenging, requiring geometric reasoning based on vanishing points.

3.2.6 Interpolating between intersection points

To allow evaluation of optical flow over a larger occluded area and not just isolated points, we sparsely extend the annotation in both inter-object and self-occlusion cases. We first identify geometrically well-defined intersections, which serve as precise anchor coordinates using the Line Intersection Method. When the occluded region involves minimal perspective transformation, we interpolate between these intersection points for annotation. The annotated area thus includes both the intersection point(s) and the interpolated points between them.

3.3. Model Selection

FlowFormer++ [13] and CCMR [6] were selected for this study due to their state-of-the-art performance in optical flow estimation, particularly under occluded regions.

FlowFormer++ [13] builds upon the transformer-based architecture of FlowFormer, through the Masked Cost Volume Autoencoding pre-training. It works by masking parts of the cost volume, a representation of pixel similarity between frames, and training the model to reconstruct the missing parts. Empirical results demonstrate its effectiveness: FlowFormer++ [13] achieves average end-point errors (EPE) of 1.07 and 1.94 on the clean and final passes of the Sintel [1] benchmark, respectively, and an F1-all score of 4.52 on the KITTI [10]-2015 test set, outperforming previous methods in these challenging scenarios.

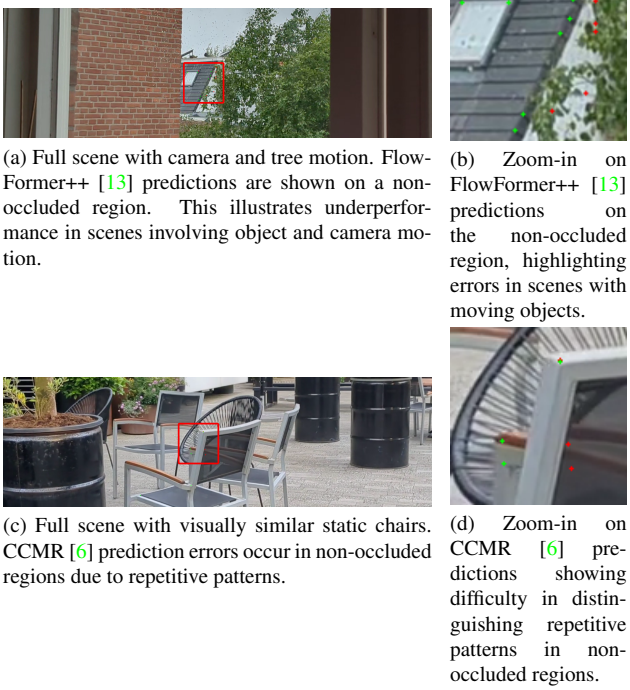
CCMR [6] addresses the shortcomings of earlier attention-based models that operated at a single scale and often produced coarse motion estimates. It uses a hierarchical two-stage attention mechanism: the first stage captures global motion context across three scales, while the second stage refines motion estimation by grouping flow informa-

Occlusion Type	FlowFormer++		CCMR	
	EPE ↓	F1-occ ↓	EPE ↓	F1-occ ↓
All-occ	18.58	64.43	8.27	22.53
Self-Occlusion	24.97	74.73	15.10	42.10
Out-of-Frame	19.48	62.10	4.00	21.05
Inter-Object	7.05	56.38	1.16	4.25

Table 1. Performance of FlowFormer++ [13] and CCMR [6] on the proposed real-world dataset, broken down by occlusion type. Metrics are the average End-Point Error (EPE) and F1-occ, namely, evaluation on occluded pixels. "All-occ" refers to all occluded pixels across the dataset. CCMR [6] outperforms FlowFormer++ [13] across all occlusion types. Self-occlusion is the most challenging, while inter-object occlusion yields the best performance for both models.

Occlusion Type	FlowFormer++		CCMR	
	EPE ↓	F1-noc ↓	EPE ↓	F1-noc ↓
All-noc	10.35	26.41	1.61	2.83

Table 2. Performance on non-occluded pixels (F1-noc) for FlowFormer++ [13] and CCMR [6]. "All-noc" refers to all non-occluded pixels in the dataset. While both models are expected to perform well, FlowFormer++ [13] underperforms significantly, suggesting sensitivity to scene confounders even outside occlusions.



(a) Full scene with camera and tree motion. FlowFormer++ [13] predictions are shown on a non-occluded region. This illustrates underperformance in scenes involving object and camera motion.

(b) Zoom-in on FlowFormer++ [13] predictions on the non-occluded region, highlighting errors in scenes with moving objects.

(c) Full scene with visually similar static chairs. CCMR [6] prediction errors occur in non-occluded regions due to repetitive patterns.

(d) Zoom-in on CCMR [6] predictions showing difficulty in distinguishing repetitive patterns in non-occluded regions.

Figure 5. Predicted flow with ground truth (green) and prediction (red) points. Left: full scenes. Right: zoom-in on the marked regions. These examples highlight typical pitfalls even in non-occluded regions due to motion or repetitive patterns.

tion across these scales. CCMR [6] achieves average end-point-errors of 1.19 and 2.14 on the clean and final passes of the Sintel [1] benchmark, respectively, and an F1-all score of 4.04 on the KITTI [10]-2015 test set.

These models were chosen for their innovative architectures and demonstrated success in handling occlusions, making them suitable for evaluating performance on datasets specifically designed to test occlusion scenarios.

4. Experiments

The evaluation aimed to compare the performance of FlowFormer++ [13] and CCMR [6] across different occlusion types. To support this, the dataset includes 22 frame pairs with sparse annotations: 95 out-of-frame, 94 inter-object (across 6 scenes each), and 95 self-occlusion annotations (across 10 scenes). The dataset covers a diverse set of 22 scenes, 9 outdoor and 13 indoor, all featuring camera motion, with 5 also containing moving objects.

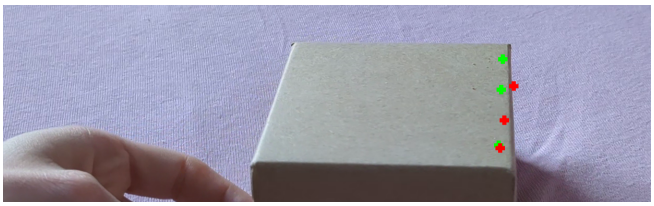
A dataset with non-occluded annotations was also created, using the same scenes and frames, with 106 annotated points. The goal of this dataset was to be able to better observe how other confounding variables contribute to the performance of the models under occluded areas.

Experiments were conducted using an NVIDIA Quadro P2000 GPU. Evaluation followed average end-point-error (EPE) and KITTI [10]'s F1 metrics. EPE measures the average Euclidean distance between predicted and ground truth optical flow vectors, while the F1(%) score captures the proportion of pixels where the endpoint error exceeds 3 pixels and 5% of the ground truth magnitude. We report F1-occ for annotated occluded pixels and F1-noc for annotated non-occluded pixels.

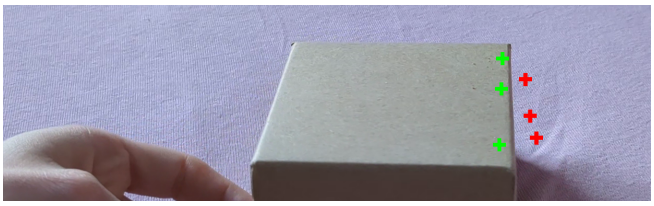
4.1. Annotation Accuracy

Given that the ground truth is manually annotated, an accuracy error margin should be considered. Annotations for non-occluded regions may be off by 1–2 pixels. For out-of-frame occlusions, a similar margin is reasonable, as annotated points remain visible in both frames. However, in inter-object and self-occlusion cases, a 2–3 pixel error margin is more likely, making F1 scores more sensitive due to the 3-pixel threshold used.

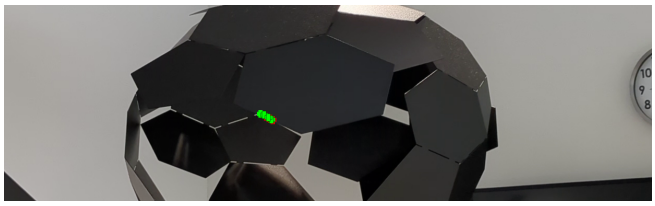
Since only a limited number of points are labelled per scene, F1 provides a more reliable indicator of the model failure frequency, particularly in the presence of outliers. In cases where the F1 score falls within the annotation uncertainty range, the average EPE becomes a dependable measure.



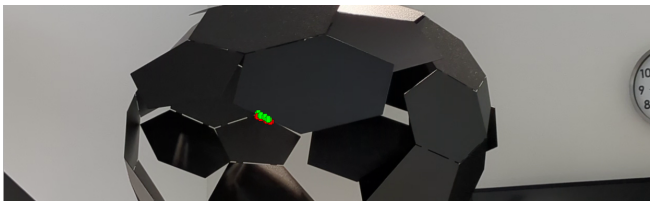
(a) CCMR [6] prediction on the annotation of the bottom-right back corner of the box, highlighting the challenge of self-occlusion under rotation.



(b) FlowFormer++ [13] prediction points on the annotation of the bottom-right back corner of the box, showing the difficulty of handling self-occlusions caused by object rotation.



(c) CCMR [6] prediction on a hexagonal region from part of a lamp, suggesting that camera translation is easier for the model to handle than object rotation.



(d) FlowFormer++ [13] prediction on a hexagonal region from part of a lamp, highlighting that camera translation is generally easier than handling rotation.

Figure 6. Predicted points for self-occlusions from CCMR [6] and FlowFormer++ [13], with ground truth in green and predictions in red. Results show that perspective transformation and parallax are challenges in self-occluded scenes.

4.2. Quantitative Results

As shown in Table 1, CCMR [6] consistently outperforms FlowFormer++ [13] across all occlusion types, in both EPE and F1-occ metrics. Inter-object occlusions yield the best results for both models, likely because they involve motion continuity and clear object boundaries. Since both objects remain in the frame, unlike out-of-frame occlusions, models don't need to hallucinate motion. These scenes also typically involve less perspective transformation than self-occlusions, and the occluded objects often have stronger visual boundaries. Additionally, inter-object occlusions may be better represented and annotated in the training data, making them more familiar to both models.

In contrast, self-occlusions perform the worst. They might be rarely annotated due to their difficulty in real-world data, which explains the weaker performance. Moreover, the parallax effect from camera translation and rotation causes appearance changes and ambiguous correspondences, making flow estimation more challenging.

An important factor in evaluating performance on occluded regions is accounting for other scene-level challenges, such as motion blur or large displacements. To quantify these effects, we also assess model performance on non-occluded regions. As shown in Table 2, CCMR [6] consistently outperforms FlowFormer++ [13] in these regions, indicating stronger generalization and robustness to challenging conditions. For completeness, we address the few scenes where CCMR [6] underperforms in Section 4.3.

Surprisingly, despite FlowFormer++ [13]'s strong benchmark performance, we expected better generalization across scenes. While it performs well in certain non-occluded areas, this may stem from overfitting to the KITTI [10] dataset, where training and testing data share similar scene features. This risk is potentially amplified by its model size, 18.2M parameters compared to CCMR [6]'s 10.8M, a reduction of over 40%. We further examine scenes with high F1-noc scores in Section 4.3.

4.3. Qualitative Results

To better understand the strengths and limitations of the models, this subsection presents a visual analysis of the optical flow predictions from FlowFormer++ [13] and CCMR [6]. We achieve this by marking the ground truth annotations and predicted points on the second frame of each image pair. We begin by comparing performance on the easier case, the non-occluded regions, taking into account scene-specific confounders. We then discuss the qualitative results in occluded scenarios.

To assess model robustness outside occluded areas, we compare scenes with 0% F1-noc scores, indicating full accuracy, with those showing non-zero scores. While non-occluded regions should be easier to estimate, many scenes still show prediction errors, though generally smaller than those in occluded areas. This comparison helps isolate failure cases caused not by occlusion but by other challenging scene factors.

FlowFormer++ [13] shows non-zero F1-noc scores in 9 out of 22 scenes, with performance particularly affected in scenes involving object or camera motion, as shown in Figure 5a. Such motion can introduce large displacements and blur, making prediction more difficult.

In contrast, CCMR [6] yields non-zero F1-noc scores in

only two scenes, one involving textureless surfaces and the other featuring repetitive structures, as seen in Figure 5c, both of which can confuse motion estimation. This suggests that CCMR [6] is more resilient to scene confounders, maintaining higher accuracy across a wider range of real-world conditions.

To further analyze performance, we next examine representative scenes for each occlusion type, highlighting where the models succeed, fail, or diverge.

4.3.1 Self-occlusion evaluation

Self-occlusion is by far the most difficult case for the models to handle, as shown in Table 1. A common pitfall arises when the occluded region is entirely absent in the second frame, leaving no visual correspondence for the model to rely on. Additionally, when a rigid object rotates, the parallax effect causes parts of the object to appear to move at different speeds in the image plane, even if the motion is uniform in 3D space, making depth-aware reasoning more difficult. An example is shown in Figures 6a for CCMR [6] and 6b for FlowFormer++ [13], where both models fail to estimate the motion of annotated corners that become occluded due to perspective transformation.

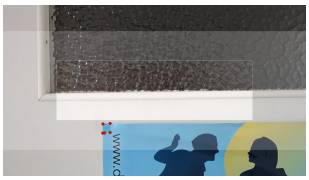
In simpler self-occlusion cases with minimal rotation, the challenge is significantly reduced. When the perspective transformation and parallax are limited, occluded regions remain close to their original positions, making the scene more predictable. This is illustrated in Figures 6c and 6d, where one polygonal edge of a lamp is occluded by another in the second frame.

In the example scenes, CCMR [6] is not affected by confounding variables, as confirmed by the non-occluded F1 scores and EPE results. This suggests that self-occlusion is the primary source of error. In contrast, FlowFormer++ [13] shows a degraded F1-noc score in Figure 6b, suggesting that object and camera motion may introduce displacements that further impact its performance.

4.3.2 Out-of-frame evaluation

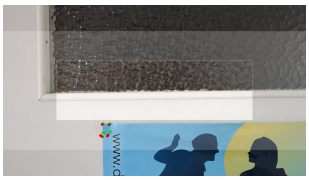
Out-of-frame occlusions are challenging because there are no visual correspondences in the second frame. While these scenes tend to perform better than self-occlusions, likely due to smaller perspective transformations, most involve only camera translations with minimal rotation, making them more manageable.

An interesting case is when an entire object moves out of frame and is no longer visible in the second image. In such situations, FlowFormer++ [13] tends to hallucinate motion, as seen in Figure 7b, despite accurately predicting the non-occluded areas in the same scene. The object, a poster, disappears completely in the second frame. Remarkably,



(a) CCMR [6] prediction on the corner of the poster, showing almost accurate reasoning of a full out-of-frame occlusion.

(c) CCMR [6] prediction on window and text corners, illustrating strong visual cues improve performance in out-of-frame occlusion.



(b) FlowFormer++ [13] prediction on the corner of the poster, showing motion hallucination when the object moves fully out of frame.

(d) FlowFormer++ [13] prediction on window and text corners, showing good performance from strong visual cues in an out-of-frame setting.

Figure 7. Predictions for out-of-frame occlusions from CCMR [6] and FlowFormer++ [13] with ground truth in green and predictions in red. Scenes with strong visual cues are easier to extrapolate, while full out-of-frame occlusions remain difficult.

CCMR [6] handles the occlusion well, its predictions stay close to ground truth despite the object vanishing entirely.

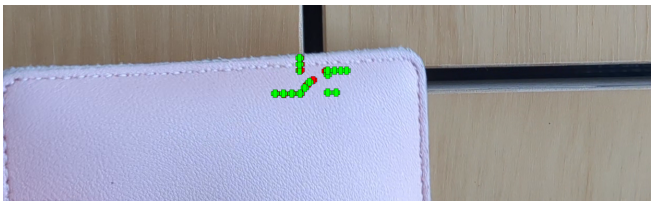
Still, out-of-frame occlusion can be manageable when visual cues are available. For instance, in Figures 7c and 7d, parts of a building move out of frame, but both models perform well. This likely comes from the building's flat, rigid structure and persistent features, which reduce ambiguity and make motion easier to extrapolate.

4.3.3 Inter-object occlusion evaluation

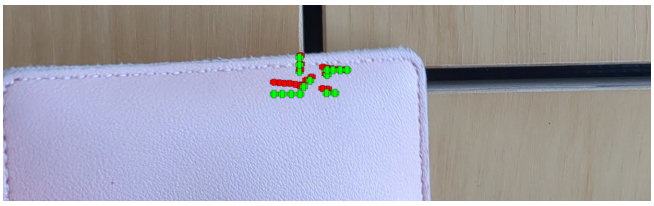
Inter-object occlusions tend to be the easiest for both models, especially for CCMR [6]. This may be because both objects remain visible, and such cases are likely well-represented in the training data.

Accurate flow in inter-object occlusion cases often depends on preserving motion boundaries, particularly when both camera and object motion are present. As noted in Section 4.3, FlowFormer++ [13] already struggles under such conditions in non-occluded areas. Its performance worsens when occluded regions are introduced, as shown in Figure 8b, where the space between wooden panels is occluded by a pink case. In contrast, CCMR [6] handles the scene more effectively.

Finally, Figures 8c and 8d show an inter-object occlusion where low motion, sharp edges, and clear texture cues make this scene relatively easy for both models. In the scene, the top of a pole occludes the background building due to camera motion.



(a) CCMR [6] prediction for inter-object occlusion with a moving pink case and background panels, suggesting the model handles combined camera and object motion effectively.



(b) FlowFormer++ [13] prediction on a scene with a moving pink case and background panels shows difficulty tracking occluded areas with other present confounders, specifically large displacements.

(c) CCMR [6] predicts the corner of the building almost perfectly, from low motion and strong visual cues.

(d) FlowFormer++ [13] performs well on static objects, as seen on the corner of the building prediction.

Figure 8. Predicted flow from CCMR [6] and FlowFormer++ [13] with ground truth (green) and prediction (red). CCMR [6] excels in inter-object occlusion, while FlowFormer++ [13] handles scenes with only camera motion better than complex scene movements.

5. Responsible Research

While occlusions can also involve humans, our dataset mostly contains images of objects. Scenes that include a foot or a hand are those where the author assisted in adding motion to objects or scenes. In some cases, people appear in the background, but they are cropped out during annotation, as they fall outside KITTI [10]'s resolution. Therefore, the evaluation targets only objects, not human subjects.

We ensure reproducibility by providing the source code of our annotation tool, along with the dataset containing the videos and selected frames upon request. The annotation process is described in detail in this paper, grouped by occlusion type, further supporting replicability.

Biases in our evaluation may have been introduced, as we annotate edges and intersections, which could skew the results toward more challenging cases. Edges often indicate depth changes, making these regions more difficult for models to estimate accurately.

6. Discussion

6.1. Contributions and insights

This work investigated how optical flow models, specifically FlowFormer++ [13] and CCMR [6], handle occlusions in real-world scenes. We introduced a new dataset along with an annotation pipeline, covering frame extraction, occlusion-specific annotation, and evaluation. Inter-object and self-occlusions were annotated using the Line Intersection Method, often extended by interpolating annotations between intersection points of occluded areas. Out-of-frame cases used a visual border that indicates the KITTI [10] resolution as a boundary to track points leaving the visible scene. To capture broader model behaviour, we also included non-occluded regions, helping distinguish occlusion-specific errors from other scene-level confounders.

Our analysis revealed that self-occlusions were the most challenging, primarily due to parallax effects and perspective transformation. Inter-object occlusions, by contrast, allowed for better performance, likely due to clear object boundaries and visibility of both objects. Out-of-frame occlusions were sensitive to context: when persistent visual cues were present, models performed well; in their absence, performance dropped due to hallucinated flow.

Among the evaluated models, CCMR [6] consistently outperformed FlowFormer++ [13] in both occluded and non-occluded regions, achieving over 65% better performance in F1-occ scores, and over 89% lower F1-noc score.

6.2. Limitations

While our dataset focuses on occlusion evaluation, confounders can still affect performance under occlusions. Although the goal is to isolate and assess limitations in occlusion handling, real-world scenes inherently include confounding factors, and removing them would result in an overly narrow and unrealistic dataset.

Our manual annotation approach relies on visible geometry, limiting what can be annotated and biasing toward harder cases, especially near depth boundaries. Extending annotations into occluded areas is only done when the perspective transformation is small. Therefore, a formal error tolerance or multi-annotator verification is still missing.

Future work includes expanding the dataset to cover more diverse scenes, and refining the annotation tool with semi-automated assistance to improve efficiency and consistency.

6.3. Conclusion

This study demonstrates that current optical flow models, specifically FlowFormer++ [13] and CCMR [6], still struggle with occlusion scenarios, particularly those involving self-occlusion and large perspective transformations. By providing an annotation pipeline, a new occlusion-focused dataset, and comparative evaluation across occlusion types, we offer insights into model weaknesses.

Our findings suggest that while CCMR [6] shows promising robustness, significant challenges remain for all models in accurately handling occluded regions. This highlights the need for continued research into improved model design for occlusion handling and more diverse real-world evaluation benchmarks.

References

[1] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 611–625, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg. 2, 4, 5

[2] Haiyang Chao, Yu Gu, and Marcello Napolitano. A survey of optical flow techniques for robotics navigation applications. *Journal of Intelligent and Robotic Systems: Theory and Applications*, 73(1-4):361 – 372, 2014. 1

[3] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazırbaş, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, 2015. 1

[4] Berthold K.P. Horn and Brian G. Schunck. Determining optical flow. *Artificial Intelligence*, 17(1-3):185–203, 1981. 2

[5] X. Huang, R. Fernandez-Rojas, and K.L. Ou. Cortical activation investigation by optical flow and wavelet analysis using near-infrared spectroscopy. volume 2016-July, page 1307 – 1312, 2016. 1

[6] Azin Jahedi, Maximilian Luz, Marc Rivinius, and Andrés Bruhn. Ccmr: High resolution optical flow estimation via coarse-to-fine context-guided motion reasoning, 2023. 1, 2, 4, 5, 6, 7, 8, 9

[7] Shihao Jiang, Dylan Campbell, Yao Lu, Hongdong Li, and Richard Hartley. Learning to estimate hidden motions with global motion aggregation. page 9752 – 9761, 2021. 1, 2

[8] Bruce D Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI'81: 7th international joint conference on Artificial intelligence*, volume 2, pages 674–679, Vancouver, Canada, Aug. 1981. 2

[9] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 4040–4048. IEEE, June 2016. 1

[10] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3061–3070, 2015. 1, 2, 3, 4, 5, 6, 8

[11] Milin P. Patel and Shankar K. Parmar. Moving object detection with moving background using optic flow. 2014. 1

[12] Xiaoyu Shi, Zhaoyang Huang, Weikang Bian, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Videoflow: Exploiting temporal cues for multi-frame optical flow estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12469–12480, 2023. 2

[13] Xiaoyu Shi, Zhaoyang Huang, Dasong Li, Manyuan Zhang, Ka Chun Cheung, Simon See, Hongwei Qin, Jifeng Dai, and Hongsheng Li. Flowformer++: Masked cost volume autoencoding for pretraining optical flow estimation, 2023. 1, 2, 4, 5, 6, 7, 8, 9

[14] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *European Conference on Computer Vision (ECCV)*, pages 402–419. Springer, 2020. 1, 2

[15] Yang Wang, Yi Yang, Zhenheng Yang, Liang Zhao, Peng Wang, and Wei Xu. Occlusion aware unsupervised learning of optical flow, 2018. 1

[16] Xiaomei Zeng, Xingcong Zhao, Xinyue Zhong, and Guangyuan Liu. A survey of micro-expression recognition methods based on lbp, optical flow and deep learning. *Neural Processing Letters*, 55(5):5995–6026, 2023. 1